

PCI10-Mbps Frame Synchronization Reed-Solomon (RSEDG) Card Hardware Definition Document

Review

June 1997



National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

PCI10-Mbps Frame Synchronization Reed-Solomon (RSEDG) Card Hardware Definition Document

August 1997

Responsible Engineer:

Reviewed by:

**J. Hui, Lead Engineer
Next Generation Systems Group
Lockheed-Martin Space Mission Systems**

**S. Linehan, Manager
Satellite Ground Systems Group
RMS Technologies, Inc.**

Approved by:

Approved by:

**P. Ghuman, Technical Lead
Next Generation Systems Group
Code 521, NASA GSFC**

**N. Speciale, Branch Head
Microelectronic Systems Branch
Code 521, NASA GSFC**

Edited by:

**L. Kane, Sr. Technical Writer
NYMA, Inc.**

**Goddard Space Flight Center
Greenbelt, Maryland**

PREFACE

This document is under the configuration management of the Microelectronic Systems Branch Configuration Control Board. Changes to this document shall be made by Documentation Change Notice, reflected in text by change bars, or by complete revision.

Requests for copies of this document, along with questions and proposed changes, should be addressed to:

**Technology Support Office
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771
(301) 286-6034**

Change Information Page

List of Effective Pages			
Page Number		Issue	
Title		Draft	
Signature Page		Draft	
iii through xiii		Draft	
1-1 through 1-7		Draft	
2-1 through 2-5		Draft	
3-1 through 3-13		Draft	
4-1 through 4-5		Draft	
5-1 through 5-4		Draft	
6-1 and 6-3		Draft	
AB-1 and AB-2		Draft	
Document History			
Document Number	Status/Issue	Publication Date	CCR Number
521-S/W-058	Review	June 1997	

DCN Control Sheet

DCN Number	Date/Time Group (Teletype Only)	Month/Year	Section(s) Affected	Initials

TABLE OF CONTENTS

Section 1: General Information

1.1	Purpose.....	1-1
1.2	Scope	1-1
1.3	General Description.....	1-1
1.4	Reference Documents.....	1-5

Section 2: Functional Description

2.1	Introduction	2-1
2.2	Theory of Operation	2-2
2.3	Modes of Operation	2-3
2.4	Functional Elements.....	2-3
2.4.1	PCI Bus Interface	2-3
2.4.2	Input/Output Interface.....	2-4
2.4.3	Parallel Integrated Frame Synchronizer	2-4
2.4.4	Reed-Solomon Error Detector/Corrector.....	2-4
2.4.5	Baseboard Interface	2-4
2.4.6	DMA Chaining RAM.....	2-4
2.4.7	Flash EEPROM.....	2-5
2.4.8	Forward Link Interface.....	2-5
2.5	Interrupts	2-5

Section 3: Memory Map and Register Definitions

3.1	Introduction	3-1
3.2	Memory Map	3-1
3.3	Memory Map Description and Register Definitions	3-1
3.3.1	Board Control Register.....	3-2
3.3.2	Status Register 0.....	3-4
3.3.3	Status Register 1.....	3-5
3.3.4	Status Register 2.....	3-7
3.3.5	Programmable Flag/Test Data Register	3-9
3.3.6	Miscellaneous Control Register	3-11
3.3.7	PIFS Registers	3-12
3.3.8	RS Internal RAM.....	3-12
3.3.9	RS Registers.....	3-12
3.3.10	RS Routing Table RAM.....	3-12
3.3.11	FIFO #1.....	3-12
3.3.12	FIFO #2.....	3-12
3.3.13	FIFO #3.....	3-12
3.3.14	FIFO #4.....	3-12
3.3.15	Forward Link FIFO	3-13
3.3.16	Flash EEPROM.....	3-13
3.3.17	Board RAM.....	3-13

TABLE OF CONTENTS (CONT'D)**Section 4: Hardware Installation**

4.1	Introduction	4-1
4.2	Hardware Elements.....	4-1
4.2.1	I/O Panel.....	4-1
4.2.2	Internal Connectors.....	4-2
4.2.2.1	Internal Connectors.....	4-2
4.2.3	Jumpers and Switches.....	4-2
4.3	Interfaces.....	4-2
4.3.1	LED.....	4-2
4.3.2	RS-422 Data Input Interface.....	4-3
4.3.3	RS-422 Forward Link Data Output Interface	4-3
4.3.4	PCI Bus Interface	4-4
4.4	Installation Guidelines.....	4-5
4.5	Environmental Requirements.....	4-5

Section 5: Operating Principles

5.1	Introduction	5-1
5.2	Preliminaries.....	5-1
5.3	Setup.....	5-1
5.3.1	Resets and Inputs	5-1
5.3.2	PCI Interface	5-1
5.3.3	Input Interface.....	5-2
5.3.4	PIFS	5-2
5.3.5	Data Routing Mode	5-2
5.3.6	RS.....	5-2
5.3.7	Interrupts	5-2
5.3.8	Programmable FIFO Flags	5-2
5.3.9	Baseboard Interface	5-3
5.4	Operation	5-3
5.4.1	Wait for Data.....	5-3
5.4.2	Initiate a Transfer.....	5-3
5.4.3	Wait for DMA Completion.....	5-4
5.4.4	Recheck for Data.....	5-4
5.4.5	Flush Remaining Data	5-4

TABLE OF CONTENTS (CONT'D)**Section & Schematic Description**

6.1	Introduction	6-1
6.2	Drawing Directory	6-1
6.3	Drawing Description	6-1
6.3.1	Top Assembly	6-1
6.3.2	Bottom Assembly.....	6-1
6.3.3	Drill Assembly.....	6-2
6.3.4	Sheet 1	6-2
6.3.5	Sheet 2	6-2
6.3.6	Sheet 3	6-2
6.3.7	Sheet 4	6-2
6.3.8	Sheet 5	6-2
6.3.9	Sheet 6	6-2
6.3.10	Sheet 7	6-2
6.3.11	Sheet 8	6-2
6.3.12	Sheet 9	6-2
6.3.13	Sheet 10.....	6-2
6.3.14	Sheet 11.....	6-2
6.3.15	Sheet 12.....	6-2
6.3.16	Sheet 13.....	6-3
6.3.17	Sheet 14.....	6-3
6.4	Program Listing Directory	6-3
6.4.1	Baseboard Interface Program (CPUSIGS.SRC).....	6-3
6.4.2	Forward Link Interface Program (P2S.SRC).....	6-3
6.4.3	Channel Routing Stack Ordering Program (STACKER.SRC).....	6-3
6.4.4	Interrupt Generation Program (INT.SRC).....	6-3
6.4.5	PCI Configuration ROM.....	6-3
Acronyms And Abbreviations		AB-1
Appendix: A - Board Layout and Schematics.....		A-1
Appendix: B - Programmable Device Listings		B-1
Appendix: C - Parts List.....		C-1
Appendix: D - Engineering Change Orders.....		D-1
Appendix: E - Test Procedures.....		E-1

TABLE OF CONTENTS (CONT'D)**Tables**

Table 3-1	Memory Map	3-1
Table 3-2	Board Control Register.....	3-2
Table 3-3	Status Register 0.....	3-4
Table 3-4	Status Register 1.....	3-5
Table 3-5	Status Register 2.....	3-7
Table 3-6	Miscellaneous Control Register	3-11
Table 4-1	PCI10FR I/O Panel Interfaces.....	4-3
Table 4-2	PCI10FR PCI Bus Connector P1 Signals	4-4
Table 6-1	PCI10FR Drawing Directory.....	6-1
Table 6-2	PCI10FR Program Listing Directory.....	6-3

Figures

Figure 1-1	Code 521 Current Generation Return-Link Data Processing System	1-2
Figure 1-2	Code 521 Next Generation PCI 10-Mbps Frame-Synchronization Reed-Solomon EDC Card	1-3
Figure 1-3	Code 521 Next Generation Desktop Satellite Data Processor.....	1-4
Figure 2-1	PCI10FR Card Functional Block Diagram.....	2-1
Figure 2-2	PCI10FR Mbps ES RSEC Assembly Board	2-3
Figure 4-1	PCI10FR I/O Panel	4-1
Figure 4-2	PCI10FR PCI Connector P1	4-2

PCI 10-MBPS FRAME-SYNCHRONIZATION REED-SOLOMON EDC CARD SUMMARY INFORMATION

FEATURES

- Performs telemetry return-link data synchronization, Reed-Solomon(RS) error detection and correction, and forward link commanding functions for a ground station.
- Works in common Peripheral Components Interface (PCI)-compliant host computers.
- Up to 10 Mbps telemetry data input rate.
- RS-422 bit-serial telemetry data input.
- Serial telemetry data loop back input direct from host computer.
- Performs all functions of a telemetry data frame synchronizer (FS) on both Consultative Committee on Space Data Systems (CCSDS) and non-CCSDS (Weather, etc.) data.
- Performs all functions of a RS error detector/corrector (EDC) on frame-synchronized data.
- Distribute data into up to four channel routing first-in-first-out (FIFO) buffers.
- Data queued up in 32-bit-wide transfer buffer for maximum PCI burst rate of 1 Gb/s.
- Interrupt-driven or polled service methods.
- Polled status read out.
- Digital ambient temperature monitoring.
- No onboard microprocessing unit (MPU), heat sinks, or jumpers.
- Switchable RS422 input/output (I/O) load resistor on a RS422 I/O bus.

PHYSICAL DESCRIPTION

- Universal voltage (5 or 3.3 V), 32-bit, 33 MHz, full-height, full length PCI Expansion Card.
- One I/O Panel Light-Emitting Diode (LED).
- One I/O Panel DB-25 jack for RS-422 data and clock signal I/O.
- On-board timecode reference clock oscillator.
- Two National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) Mission Operations and Data Systems Directorate (MO&DSD) Data Systems Technology Division (DSTD) Microelectronic Systems Branch (MSB) (Code 521) custom Application-Specific Integrated Circuits (ASICs):
 - Parallel, Integrated Frame Synchronizer (PIFS).
 - Reed-Solomon Error Detector/Corrector (RSEDC).
- Commercial Off-the-Shelf (COTS) PCI Local Bus interface ASIC.
- Seven Programmable Logic Devices (PLD).

SECTION 1

GENERAL INFORMATION

1.1 PURPOSE

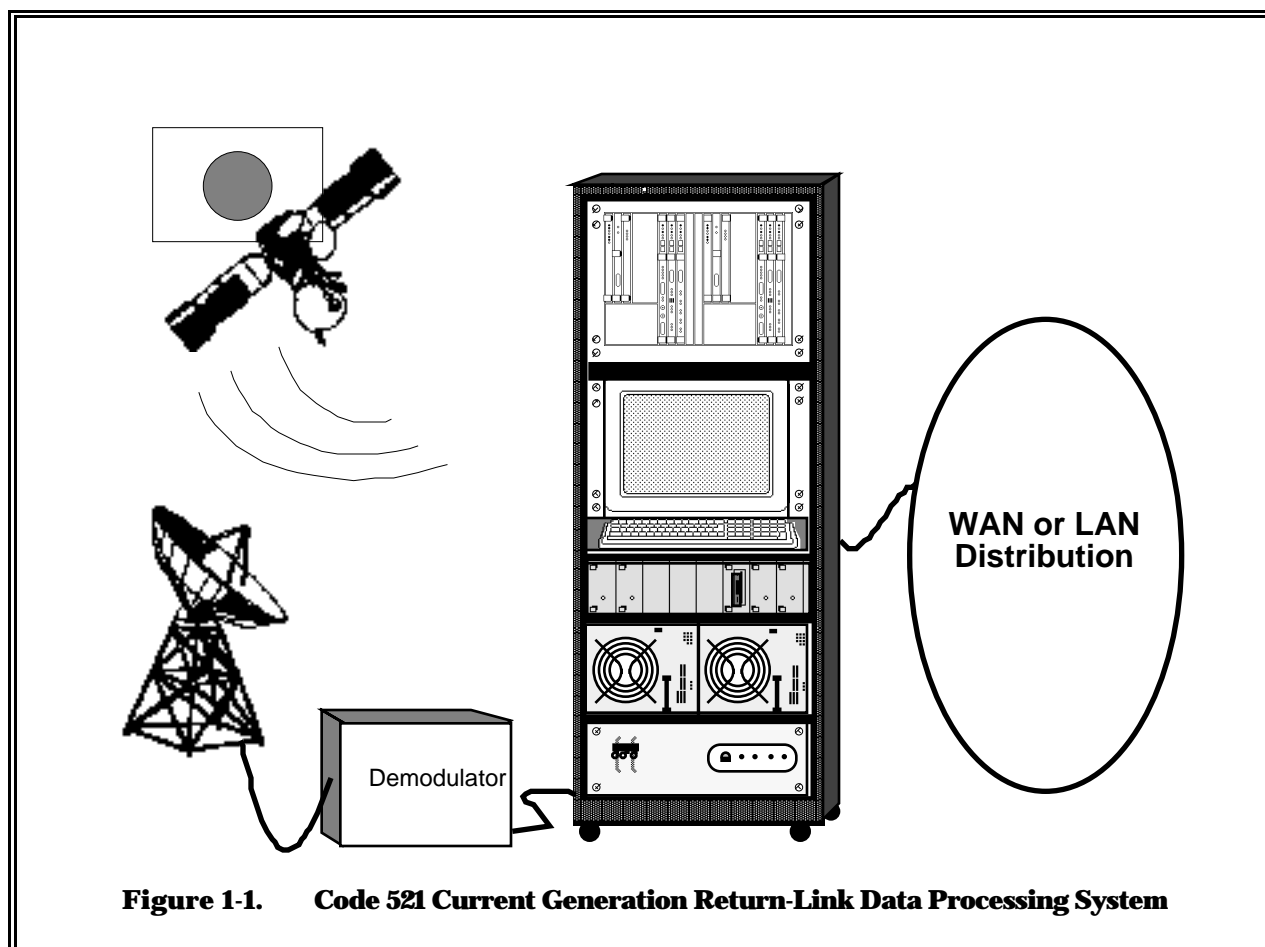
This document describes the Peripheral Component Interface (PCI) 10-Mbps Frame Synchronization (FS) Reed-Solomon Error Detection Correction (RSEDC) Card, part number G1527449, developed by the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC) Code 521 Next Generation Systems (NGS) group.

1.2 SCOPE

This document provides a functional description, memory map, register models, processing scenario, and schematic description for the PCI10FR. It is assumed that the reader is familiar with the specification and operation of: PCI Local Bus, Systems and Expansion Cards; V3 Corporation V962PBC PCI Local Bus Bridge Application Specific Integrated Circuits (ASIC); Consultative Committee on Space Data Systems (CCSDS) Packet Telemetry and Advanced Orbiting Systems (AOS) Services; NASA Communications (Nascom) data signals; telemetry return-link data processing functions including frame synchronization, and Reed-Solomon error detection and correction; and NASA GSFC Code 521 ASICs including the Parallel Integrated Frame Synchronizer (PIFS), and RS.

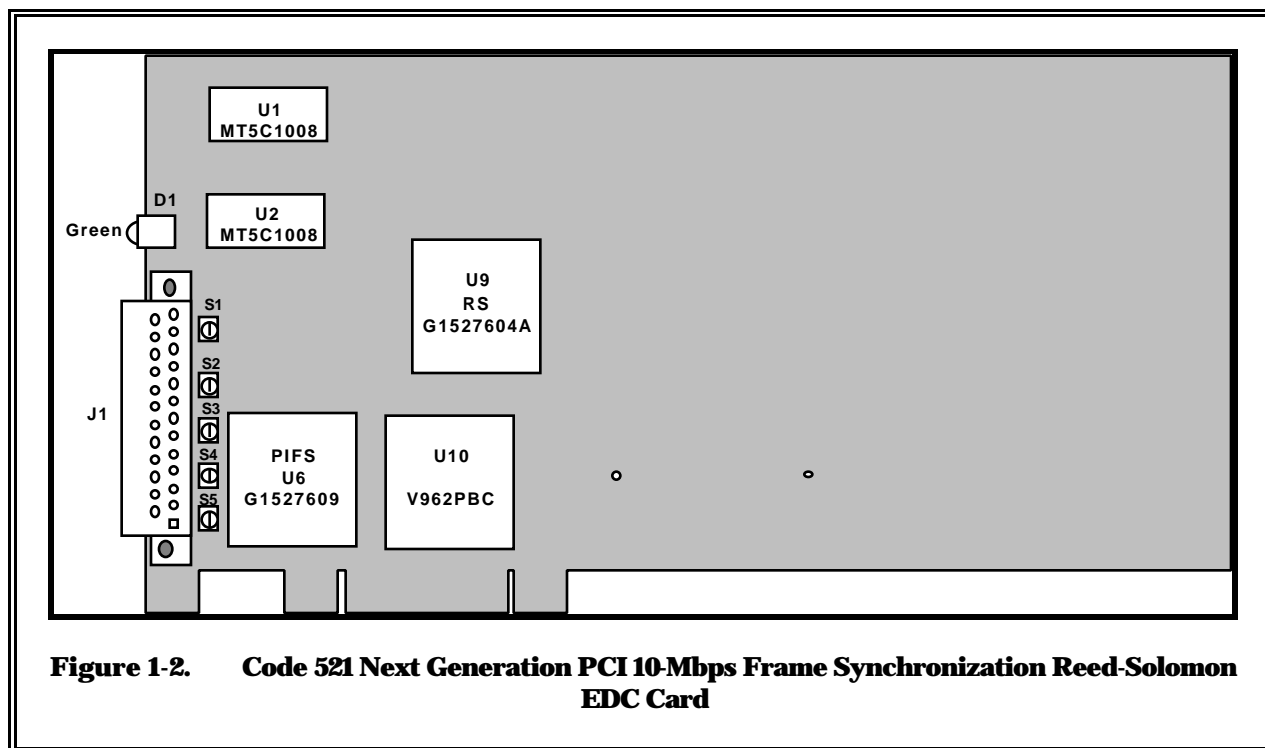
1.3 GENERAL DESCRIPTION

The current generation of Code 521-developed telemetry return-link data processing systems requires a number of Versa-Module Euroboard (VME) cards, custom ASICs, and embedded Microprocessing Units (MPUs) to perform high-speed (> 2 Mbps) real-time return-link data processing. In particular, frame synchronization at 50 Mbps and below is performed by a VME card containing a Code 521 Complimentary Metal-Oxide Semiconductor (CMOS) FS ASIC and support circuitry; frame synchronization above 50 Mbps is performed by a VME card containing a Code 521 gallium-arsenide (GaAs) FS ASIC and support circuitry; Reed-Solomon error detection and correction at up to 528 Mbps is performed by a VME card containing a Code 521 CMOS RS ASIC and support circuitry; and CCSDS Packet Telemetry and AOS Service processing at up to 50 Mbps is performed by a VME card containing two Code 521 CMOS telemetry data pipeline ASICs, three Motorola 68040 MPUs, and support circuitry. See Figure 1-1.



Continuing the evolution of telemetry processing components towards smaller/cheaper/faster by applying the current state-of-the-art in microelectronic technology, the Code 521 Next Generation Systems group recently parallelized and integrated the FS function into a CMOS ASICs capable of up to 528 Mbps (matching the RS). The PCI10FR is the first card for the next generation return-link data processing prototype, the combination of the two return link processing functions and the forward link function into a single board-level product. It is designed to operate from 0 to 10 Mbps. It is the first application of the Code 521 PIFS ASIC at 10 Mbps.

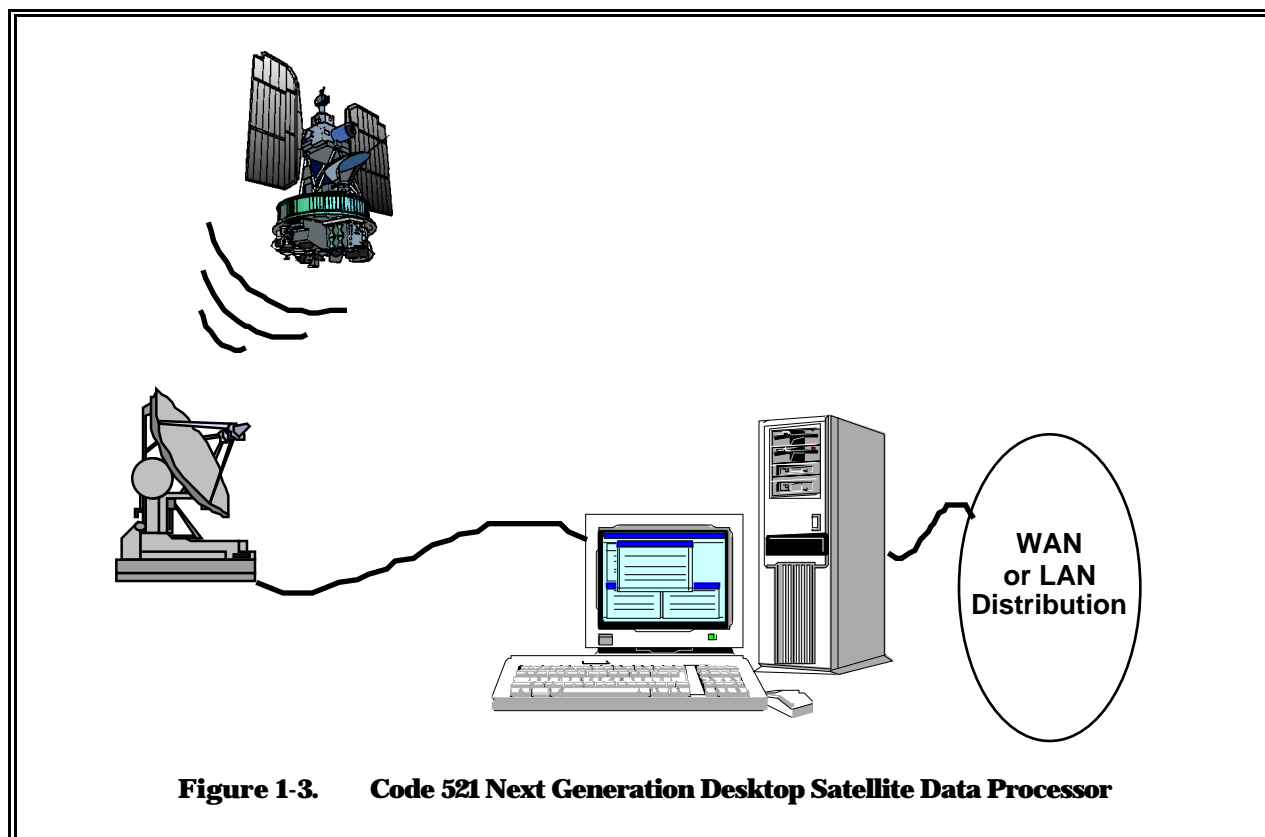
The PCI10FR is a fully Plug and Play (P&P)-compliant PCI Expansion Card capable of operating in all PCI host computers: International Business Machines (IBM)-compatibles, Macintosh-compatibles, Digital Equipment Corporation (DEC) stations, Sun SPARC stations, etc. See Figure 1-2. Thus the expense, complexity, and possible concerns about the aging VME platform are eliminated.



Serial telemetry return-link data is most commonly the output product of a bit-synchronizer, either in local receiver hardware or at Nascom; it could also be generated by test hardware or played back from storage. This enters the PIFS directly through the connector on the input/output (I/O) Panel. The host computer can inject data at relatively low rates directly into the PIFS via the PCI, so telemetry data can also be generated by the host or obtained over a network.

The telemetry data accumulates in first-in-first-out (FIFO) memory buffers for driver software running on the host computer to transfer out over the PCI bus, generally into main memory, where it is further processed or transferred to another PCI card, into storage, or out across a network.

Figure 1-3 shows how the PCI10FR fits into the Next Generation Desktop Satellite Data Processor (DSDP), the logical future of telemetry processing.



The PIFS performs CCSDS as well as custom frame synchronization, Cyclic Redundancy Code (CRC) / Bit Transition Density (BTD) decoding, CCSDS Day-Segmented Timecode generation, and frame annotation. Data from the PIFS can be input to the RS, which would be the case for most CCSDS data. The RS performs Reed-Solomon error detection and correction and frame quality annotation. Data from the RS is output via PCI. The host also has access to all of the processing status and data quality information needed for real-time monitoring.

Support for the PCI10FR on particular PCI platforms depends not on the hardware but driver software development. Driver software is currently being developed for the Intel Pentium/Microsoft Windows-NT and DEC Alpha/DEC Ultrix platforms.

Physically, the PCI10FR is a universal voltage (5 or 3.3 V), 32-bit, 33 MHz, full-height, full length PCI Expansion Card. Its I/O Panel contains one host-operated light-emitting-diode (LED), and one DB-25 jack for RS-422 data and clock signal input and RS-422 forward link data and clock output signal. The card contains an on-board timecode reference clock signal oscillator, two Code 521 custom ASICs (PIFS, RS), a COTS PCI/Local Bus Bridge ASIC, and seven Programmable Logic Devices (PLDs). See Figure 1-2.

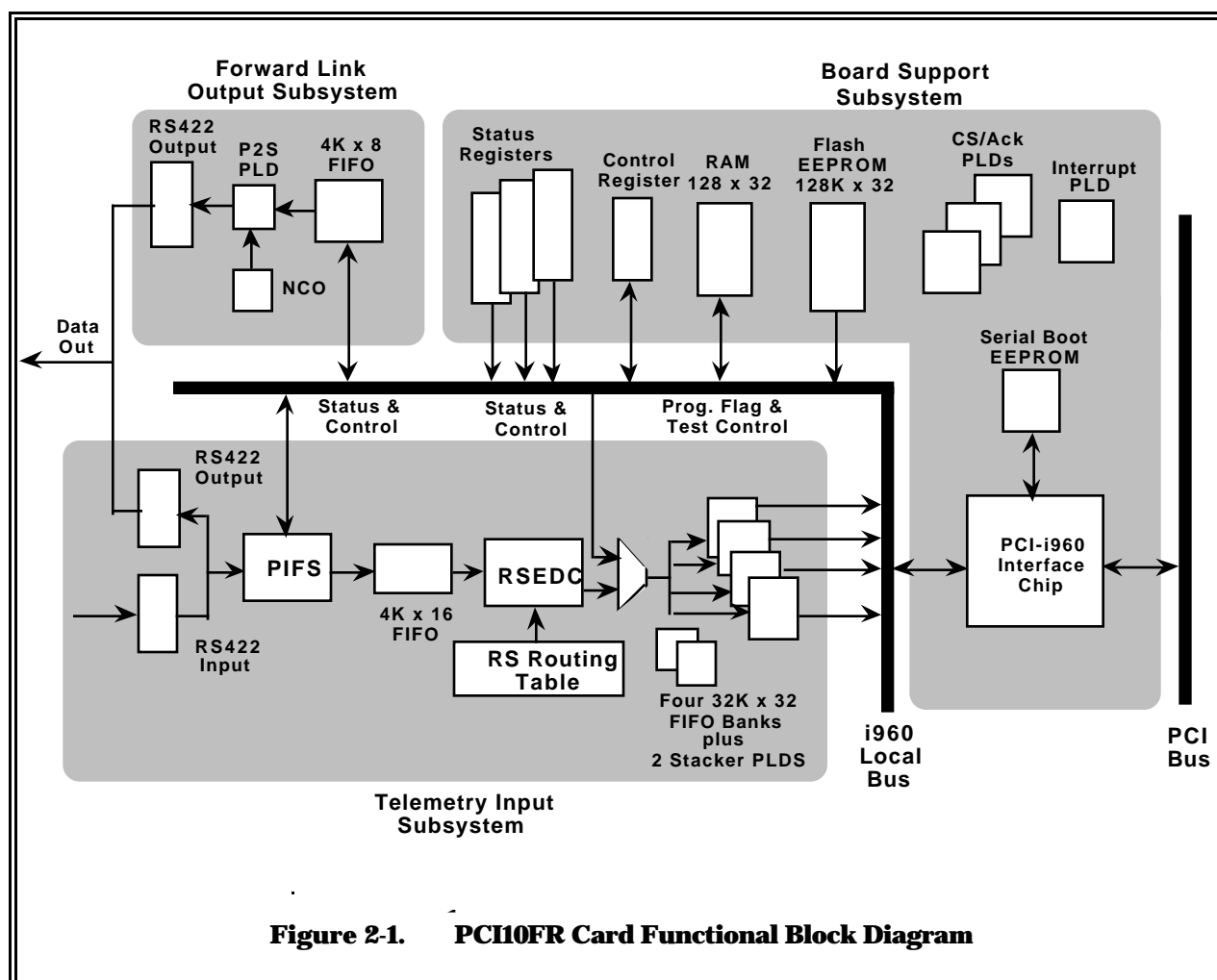
1.4 REFERENCE DOCUMENTS

- Packet Telemetry. CCSDS 102.0-B-3, CCSDS, Annapolis MD.
- Advanced Orbiting Systems, Networks, and Data Links: Architectural Specification. CCSDS 701.0-B-2, CCSDS, Annapolis MD.
- Parallel Integrated Frame Synchronizer Chip. 521-ASIC-023, NASA GSFC, Greenbelt MD.
- Microelectronic Systems Branch Application-Specific Integrated Circuits (ASIC) Components Document. 521-SPEC-002, NASA GSFC, Greenbelt MD.
- PCI Local Bus Specification Revision 2.1. PCI Special Interest Group, Portland OR.
- VxxxPBC User's Manual Revision 2.0. V3 Semiconductor Corp., Santa Clara CA.
- AT24C01A/2/4/8/16 2-Wire Serial CMOS E²PROM Data Sheet. ATMEL, San Jose CA.
- GAL26CV12-7 High Performance EE CMOS Programmable Logic Data Sheet. Lattice Semiconductor Corp., Hillsboro, OR
- IDT72241 CMOS Sync FIFO Data Sheet. Integrated Device Technology, Santa Clara CA.

SECTION 2 FUNCTIONAL DESCRIPTION

2.1 INTRODUCTION

The PCI10FR performs the satellite telemetry return-link data processing functions in real-time up to 10 Mbps. Functions include frame synchronization, Reed-Solomon error detection and correction, and forward link serial output interface, all on a single industry-standard PCI Expansion Card. In a typical scenario, a standard serial data stream is connected through the I/O Panel, processed in data flow-fashion through the Code 521 custom ASICs, and deposited in a number of FIFO memory buffers for software running on the PCI host computer to transfer elsewhere; see Figure 2-1. Weather Satellite Data and other formats can be frame-synchronized and routed through RS chip without RS decoding operation performed. Internet and other low-rate computer-accessible data sources can be injected directly by the host via PCI.



The following section provides a functional description of the PCI10FR.

2.2 THEORY OF OPERATION

The PCI10FR functions as a PCI slave device to the host computer (master). It is seen by application software as an area of memory which can be written (setup, control, data input) and read (status, data output); to see the memory map, refer to Table 3-1. The PCI10FR has no on-board MPU, so software running on the host is responsible for setup, control, monitoring, and transferring out the processed data. The card can generate interrupts to the host to request service, or its status can be polled; overall function is the same in either case.

Upon host initialization, the host operating system and driver software interact with the PCI10FR as specified by the PCI Specification to configure the device and the low-level host interface. Eventually, application software sets up the card for a "data session." Setup information ultimately comes from the user; the host sets up each of the PCI10FR components (described in §5.3) by writing to the appropriate areas of memory. The system is ready to accept telemetry data as soon as setup is complete.

Telemetry data commonly enters the card in the form of a serial bit stream through connectors on the I/O Panel. It can also be injected by the host via; this could be used for Internet, test, or other very low rate data. The data goes directly into the PIFS through the forward link local loop back connection. Output from the PIFS goes through a data FIFO to the RS and output from RS goes in up to four channel data routing FIFO buffers, per host setup.

During a data session, the host may want to obtain the status of the immediate data flow operation on the PCI10FR and other hardware conditions such as ambient temperature. This is accomplished by writing the control registers and reading the status register.

As data enters the four channel routing data FIFOs and command is output through the forward link, the fullness of each FIFOs can be read from three status buffers, along whether the FIFO receiving data from the PIFS has overflowed. These buffers can be polled by the host or generate interrupts to the host.

Most of the interrupt conditions tell the host that at least some known quantity of data is ready to be transferred out. The host typically transfers the data to host memory. To do this, the host sets up the Direct Memory Access (DMA) in the PCI Bridge Chip (PBC), which is responsible for reading the data from the channel routing FIFO across the PCI bus to the specified destination. The host is responsible for prioritizing the order in which the channel routing data FIFOs are serviced and the amount of data buffered before the need for service is indicated. Other processes running on the host typically transfer the data from host memory to storage or to another host across a network and might also perform additional data processing.

At the end of a data session, some data will likely remain in the channel routing data FIFOs; flushing these is as straightforward as a normal data transfer

After flushing, the PCI10FR may be set up for a new session, or depending on how it was previously set up, may be ready to begin another session immediately.

2.3 MODES OF OPERATION

The PCI10FR has only one mode of operation; this is what was described in §2.2.

2.4 FUNCTIONAL ELEMENTS

The functional elements of the PCI10FR consist of: the PCI Bus Interface, the I/O Interface, the Parallel Integrated Frame Synchronizer, the Reed-Solomon Error Detector/Corrector, the Baseboard Interface, DMA Chaining RAM, Flash EEPROM, and the Forward Link Interface.

2.4.1 PCI BUS INTERFACE

The PCI10FR conforms to the industry-standard PCI Local Bus Specification Revision 2.1; see §1.4 for the complete reference. It is a full-height, full-length, 33 MHz, 32-bit data, universal-voltage PCI expansion device. PCI is "Plug and Play", so the PCI10FR should function in any PCI-compliant system for which driver software has been provided. During a data burst, the PCI bus is capable of transferring one 32-bit word every 33 MHz clock cycle for a throughput of over 1 Gb/s. It receives the 33 MHz PCI bus clock from the host computer.

The local bus is a 33 MHz, fully-synchronous, 32-bit de-multiplexed address and data bus. It has the signaling and timing characteristics necessary to interface directly with the Intel i960Cx/Hx series of microprocessors, although the PCI10FR itself uses no MPU. It requires the local bus clock.

Connectivity is provided by the V962PBC PCI Bus Bridge Chip from V3 Corporation. It provides dual, chainable DMA engines (will be available when the C3 or later version of V962PBC is installed), and the PCI10FR contains 512 Kbytes of 32-bit-wide local memory to hold the chaining list. This is more than the chaining list requires, so the additional memory may be utilized by the host as desired. The PBC has been wired to generate a PCI interrupt from the Board Controller. The PBC has been wired to automatically load its configuration-space from an on-board ROM. The ROM is in-circuit re-programmable through the PBC; more information is available from the reference documents listed in §1.4. See Figure 2-2 for the PCI10 Mbps ES RSEC Assembly Board.

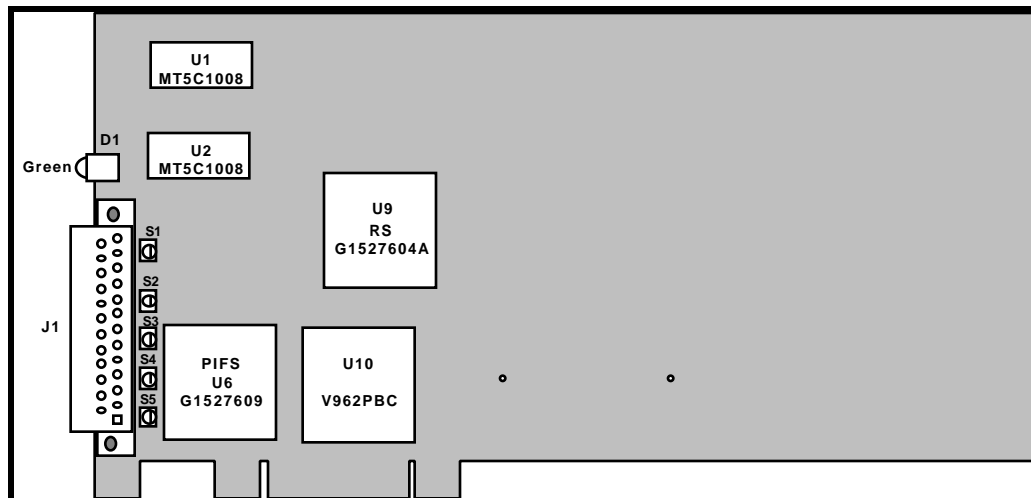


Figure 2-2. PCI10FR Mbps ES RSEC Assembly Board

2.4.2 INPUT/OUTPUT INTERFACE

The I/O Panel contains one DB-25 jack for low-rate (10 Mbps) differential RS-422 data and clock signals. Past experience with the circuits indicates that up to 25 Mbps is often possible. The low-rate telemetry inputs and the forward link output are terminated to 120W. Low-rate data enters the PIFS directly in serial form.

2.4.3 PARALLEL INTEGRATED FRAME SYNCHRONIZER

CCSDS and custom frame synchronization, Non-Return-to-Zero (NRZ) decoding, CRC checking, BTD decoding, CCSDS Day-Segmented Timecode generation, Weather Data correlation (up to 64 bits), associated frame annotation, and programmed telemetry data input is provided by the PIFS; see Figure 2-1. It runs off the 33 MHz processing clock and is theoretically capable of operating up to 66 MHz. It can input up to one byte per clock cycle and can output up to two bytes per clock cycle through a 16-bit output port. It has thirty-two 32-bit registers for control and status; more information is available from the reference documents listed in §1.4.

2.4.4 REED-SOLOMON ERROR DETECTOR/CORRECTOR

Reed-Solomon error detection and correction and associated frame quality annotation is provided by the RS. It can handle up to 8 interleaves of RS(255,223)-encoded data as well as the RS(10,6)-encoded data sometimes used for CCSDS frame headers; see Figure 2-1. It runs off the 33 MHz processing clock and is theoretically capable of operating up to 66 MHz. It can input and output one or two bytes at up to one-half the processing clock frequency through 16-bit input and output ports. It has forty 16-bit registers for control and status accessible on 32-bit boundaries; more information is available from the reference documents listed in §1.4.

2.4.5 BASEBOARD INTERFACE

The Baseboard Interface is responsible for a number of functions, the most complex of which is controlling and generating the interrupt from several possible sources. It also provides miscellaneous board control functions, decodes the major address spaces and the DMA-Chaining Memory space, Flash EEPROM Memory space, and status registers. It consists of a number of programmable logic devices running off the 33 MHz local bus clock and has five 32-bit registers. The first is the Board Control register, which allows the host to reset the PIFS chip, PIFS-RS FIFO, and RS chip, reset, change stack mode and stack order, and switch programmable flag load mode of the channel routing FIFO buffers, reset all FIFO stackers, select forward link output, toggle the LED, disable channel routing FIFO interrupts, program clock speed of the forward link output, and control of the forward link FIFO. The second register is the Miscellaneous Control register, which provides control signals for temperature readout, test points for initial testing, and masking of the forward link interrupt.

There are three status registers to provide the status of all FIFO's. The first status register contains the full flags and the empty flags of the channel routing FIFO's. The second status register contains the almost full flags and the almost empty flags of these FIFO's. And the third status register contains the status flags of the PIFS-RS FIFO, the state bits of the controller and the forward link parallel to serial converter, the clock, data, and error status of the forward link parallel to serial converter, forward link FIFO flag status, temperature sensor data, and control signals of the local bus.

2.4.6 DMA CHAINING RAM

In order to provide the DMA chaining capability for the PCI bridging chip, there is a block of 128K x 32-bit static RAM located on the local bus. There is more than enough of the memory space to be used by the end user for other applications.

2.4.7 FLASH EEPROM

A block of 128K x 32-bit flash EEPROM is provided for holding tables that are required during power up.

2.4.8 FORWARD LINK INTERFACE

The forward Link Interface is a simple byte wide circuit to serialize data into a bit stream and have it output through a RS422 port. An internal programmable bit rate clock or an external clock can be used for clocking out serial data bits.

2.5 INTERRUPTS

A single PCI device such as the PCI10FR is allowed by the PCI Specification to generate only a single interrupt. The source of the interrupt will be one or more of the interrupt conditions indicated in the Status Buffers shown in §3.3.2.

SECTION 3 MEMORY MAP AND REGISTER DEFINITIONS

3.1 INTRODUCTION

This section provides the memory map for the PCI10FR Card including device descriptions and bit definitions.

3.2 MEMORY MAP

Table 3-1. Memory Map

Address	Device	Description
xx00 0000	Board Control Register	Section 3.3.1
xx10 0000	Status Register 0	Section 3.3.2
xx12 0000	Status Register 1	Section 3.3.3
xx14 0000	Status Register 2	Section 3.3.4
xx28 0000	Programmable Flag/Test Data Register	Section 3.3.5
xx30 0000	Miscellaneous Control Register	Section 3.3.6
xx50 0000	PIFS Registers	PIFS Doc
xx68 0000	RS Internal RAM	RS Doc
xx70 0000	RS Registers	RS Doc
xx88 0000	RS Routing Table RAM	Section 3.3.10
xx90 0000	FIFO #1	Section 3.3.11
xxA0 0000	FIFO #2	Section 3.3.12
xxB0 0000	FIFO #3	Section 3.3.13
xxC0 0000	FIFO #4	Section 3.3.14
xxD0 0000	Forward Link FIFO	Section 3.3.15
xxE8 0000	Flash EEPROM	Section 3.3.16
xxF0 0000	Board RAM	Section 3.3.17

3.3 MEMORY MAP DESCRIPTION AND REGISTER DEFINITIONS

This section provides all register models for all registers listed in the memory map and defines the function of all other memory map addresses.

3.3.1 BOARD CONTROL REGISTER**Table 3-2. Board Control Register**

Reg.	Name	Board Control Register	
	Description	This is the board-level control register for the PCI10FR card.	
	Address	\$xx00 0000	
	Port Size	longword (32 bits)	
	Reset Value	0000 0000 0000 0000 0000 0000 0000 0000 (binary)	
Bit(s)	Name	R/W	Definition
	RSTPIFS	R/W	0: Reset PIFS chip 1: PIFS chip not reset
30	RSTPRF	R/W	0: Reset PIFS-RS FIFO 1: PIFS-RS FIFO not reset
29	RSTRS	R/W	0: Reset RS chip 1: RS chip not reset
28	STACKMODE	R/W	0: Pack frames (1st byte of next frames follows last byte of previous frame) 1: Pad frames (1st byte of each frame starts at a longword address)
27	STACKORDER	R/W	0: big-endian (uub-umb-lmb-llb) 1: little-endian (llb-lmb-umb-uub)
26	RSTF1	R/W	0: Reset FIFO #1 1: FIFO #1 not in reset
25	F1LD	R/W	0: FIFO #1 in programmable flag load mode 1: FIFO #1 in normal FIFO mode
24	RSTF2	R/W	0: Reset FIFO #2 1: FIFO #2 not in reset
23	F2LD	R/W	0: FIFO #2 in programmable flag load mode 1: FIFO #2 in normal FIFO mode
22	RSTF3	R/W	0: Reset FIFO #3 1: FIFO #3 not in reset
21	F3LD	R/W	0: FIFO #3 in programmable flag load mode 1: FIFO #3 in normal FIFO mode
20	RSTF4	R/W	0: Reset FIFO #4 1: FIFO #4 not in reset
19	F4LD	R/W	0: FIFO #4 in programmable flag load mode 1: FIFO #4 in normal FIFO mode
18	RSTSTACKERS	R/W	0: Reset all FIFO Stackers 1: FIFO Stackers not reset

17	TXSEL	R/W	0: RS-422 output = Forward Link output 1: RS-422 output = RS-422 Input (facilitates multiple PCI10FR cards processing one telemetry input stream)																		
16	LEDBIT	R/W	0: Back-panel LED on 1: Back-panel LED off																		
15	F1INTCTRL	R/W	0: FIFO #1 interrupt disabled 1: FIFO #1 interrupt enabled																		
14	F2INTCTRL	R/W	0: FIFO #2 interrupt disabled 1: FIFO #2 interrupt enabled																		
13	F3INTCTRL	R/W	FIFO #3 interrupt control 0: Disabled 1: Enabled																		
12	F4INTCTRL	R/W	0: FIFO #4 interrupt disabled 1: FIFO #4 interrupt enabled																		
11..10	NCOMC<1..0>	R/W	These 2 bits select one of four NCOCLK frequencies: <table><tr><td><u>NCOMC<1..0></u></td><td><u>Frequency Selected</u></td></tr><tr><td>00</td><td>50 MHz</td></tr><tr><td>01</td><td>55 MHz</td></tr><tr><td>10</td><td>Serially programmed frequency</td></tr><tr><td>11</td><td>60 MHz</td></tr></table>	<u>NCOMC<1..0></u>	<u>Frequency Selected</u>	00	50 MHz	01	55 MHz	10	Serially programmed frequency	11	60 MHz								
<u>NCOMC<1..0></u>	<u>Frequency Selected</u>																				
00	50 MHz																				
01	55 MHz																				
10	Serially programmed frequency																				
11	60 MHz																				
9	NCOSDATA	R/W	This is the Serial Data signal to the NCO. Software sequences NCOSDATA, NCOWREN, and NCOSCLK to serially program the NCO.																		
8	NCOWREN	R/W	This is the Write Enable signal to the NCO. Software sequences NCOSDATA, NCOWREN, and NCOSCLK to serially program the NCO.																		
7	NCOSCLK	R/W	This is the Serial CLock signal to the NCO. Software sequences NCOSDATA, NCOWREN, and NCOSCLK to serially program the NCO.																		
6	FLTESTCLK	R/W	When selected via CLKDIVMODE<2..0>, this bit is the serial clock input for the Forward Link Subsystem. Software can toggle this bit and observe P2SCLK and P2SDAT to test the Forward Link Subsystem.																		
5..3	CLKDIVMODE<2..0>	R/W	These 3 bits select one of 8 clock sources for the Forward Link Subsystem: <table><tr><td><u>CLKDIVMODE<1..0></u></td><td><u>Clock Selected</u></td></tr><tr><td>000</td><td>FLTESTCLK (bit 7)</td></tr><tr><td>001</td><td>NCOCLK / 16</td></tr><tr><td>010</td><td>NCOCLK / 256</td></tr><tr><td>011</td><td>NCOCLK / 4096</td></tr><tr><td>100</td><td>NCOCLK / 65 K</td></tr><tr><td>101</td><td>NCOCLK / 1 M</td></tr><tr><td>110</td><td>NCOCLK / 16 M</td></tr><tr><td>111</td><td>Externally-supplied Clock</td></tr></table>	<u>CLKDIVMODE<1..0></u>	<u>Clock Selected</u>	000	FLTESTCLK (bit 7)	001	NCOCLK / 16	010	NCOCLK / 256	011	NCOCLK / 4096	100	NCOCLK / 65 K	101	NCOCLK / 1 M	110	NCOCLK / 16 M	111	Externally-supplied Clock
<u>CLKDIVMODE<1..0></u>	<u>Clock Selected</u>																				
000	FLTESTCLK (bit 7)																				
001	NCOCLK / 16																				
010	NCOCLK / 256																				
011	NCOCLK / 4096																				
100	NCOCLK / 65 K																				
101	NCOCLK / 1 M																				
110	NCOCLK / 16 M																				
111	Externally-supplied Clock																				

2	RSTFLF	R/W	0: Reset Forward Link FIFO 1: Forward Link FIFO not in reset
1	FLLD	R/W	0: Forward Link FIFO in programmable flag load mode 1: Forward Link FIFO in normal FIFO mode
0	RSTGOFL	R/W	0: Forward Link Parallel to Serial Converter in reset 1: Forward Link Parallel to Serial Converter enabled/running

3.3.2 STATUS REGISTER 0

Table 3-3. Status Register 0

Status Register 0			
Reg	Name	Status Register 0	
	Description	This is the 1st of 3 board-level status registers on the PCI10FR card.	
	Address	\$xx10 0000	
	Port Size	longword (32 bits)	
	Reset Value	1111 1111 1111 1111 0000 0000 0000 0000 (binary) FFFF0000 (hex)	
Bit(s)	Name	R/W	Definition
31..28	F4FF<3..0>	R/O	These bits are the full flags for FIFO #4. 0: FIFO is full 1: FIFO is not full
27..24	F3FF<3..0>	R/O	These bits are the full flags for FIFO #3. 0: FIFO is full 1: FIFO is not full
23..20	F2FF<3..0>	R/O	These bits are the full flags for FIFO #2. 0: FIFO is full 1: FIFO is not full
19..16	F1FF<3..0>	R/O	These bits are the full flags for FIFO #1. 0: FIFO is full 1: FIFO is not full
15..12	F4EF<3..0>	R/O	These bits are the empty flags for FIFO #4. 0: FIFO is empty 1: FIFO is not empty

11..8	F3EF<3..0>	R/O	These bits are the empty flags for FIFO #3. 0: FIFO is empty 1: FIFO is not empty
7..4	F2EF<3..0>	R/O	These bits are the empty flags for FIFO #2. 0: FIFO is empty 1: FIFO is not empty
3..0	F1EF<3..0>	R/O	These bits are the empty flags for FIFO #1. 0: FIFO is empty 1: FIFO is not empty

3.3.3 STATUS REGISTER 1

Table 3-4. Status Register 1

Reg	Name	Status Register 1	
	Description	This is the 2nd of 3 board-level status registers on the PCI10FR card.	
	Address	\$xx12 0000	
	Port Size	longword (32 bits)	
	Reset Value	1111 1111 1111 1111 0000 0000 0000 0000 (binary) FFFF0000 (hex)	
Bit(s)	Name	R/W	Definition
31..28	F4PAF<3..0>	R/O	These bits are the programmable almost full flags for FIFO #4. 0: FIFO is almost full 1: FIFO is not almost full
27..24	F3PAF<3..0>	R/O	These bits are the programmable almost full flags for FIFO #3. 0: FIFO is almost full 1: FIFO is not almost full
23..20	F2PAF<3..0>	R/O	These bits are the programmable almost full flags for FIFO #2. 0: FIFO is almost full 1: FIFO is not almost full
19..16	F1PAF<3..0>	R/O	These bits are the programmable almost full flags for FIFO #1. 0: FIFO is almost full 1: FIFO is not almost full
15..12	F4PAE<3..0>	R/O	These bits are the programmable almost empty flags for FIFO #4. 0: FIFO is almost empty 1: FIFO is not almost empty

11..8	F3PAE<3..0>	R/O	These bits are the programmable almost empty flags for FIFO #3. 0: FIFO is almost empty 1: FIFO is not almost empty
7..4	F2PAE<3..0>	R/O	These bits are the programmable almost empty flags for FIFO #2. 0: FIFO is almost empty 1: FIFO is not almost empty
3..0	F1PAE<3..0>	R/O	These bits are the programmable almost empty flags for FIFO #1. 0: FIFO is almost empty 1: FIFO is not almost empty

3.3.4 STATUS REGISTER 2**Table 3-5. Status Register 2**

Reg	Name	Status Register #3	
	Description	This is the 3rd of 3 board-level status registers on the PCI10FR card.	
	Address	\$xx14 0000	
	Port Size	longword (32 bits)	
	Reset Value	1111 0000 0000 0000 0001 1100 0xx1 (binary) F00001CX (hex)	
Bit(s)	Name	R/W	Definition
31..30	PRFF<1..0>	R/O	PIFS-RS FIFO full flags. 0: FIFO is full 1: FIFO is not full
29..28	PRPAF<1..0>	R/O	These bits are the programmable almost full flags for the PIFS-RS FIFO. There is no hardware support for adjusting these bits from their default value. 0: FIFO is almost full 1: FIFO is not almost full
27..26	PRPAE<1..0>	R/O	These bits are the programmable almost empty flags for the PIFS-RS FIFO. There is no hardware support for adjusting these bits from their default value. 0: FIFO is almost empty 1: FIFO is not almost empty
25..24	PREF<1..0>	R/O	PIFS-RS FIFO empty flags. 0: FIFO is empty 1: FIFO is not empty
23..22	F4STATES<1..0>	R/O	These are the state machine bits for FIFO #4's byte to longword stacker. These bits in conjunction with STACKORDER indicate which byte will be written next.
21..20	F3STATES<1..0>	R/O	These are the state machine bits for FIFO #3's byte to longword stacker. These bits in conjunction with STACKORDER indicate which byte will be written next.
19..18	F2STATES<1..0>	R/O	These are the state machine bits for FIFO #2's byte to longword stacker. These bits in conjunction with STACKORDER indicate which byte will be written next.

17..16	F1STATES<1..0>	R/O	These are the state machine bits for FIFO #1's byte to longword stacker. These bits in conjunction with STACKORDER indicate which byte will be written next.
15..11	P2SSTATES<4..0>	R/O	These are the state machine bits for the parallel to serial converter.
10	P2SCLK	R/O	This bit indicated the current value of the P2S CLK bit.
9	P2SDAT	R/O	This bit indicated the current value of the P2S DAT bit.
8	P2SERR	R/O	P2S underflow error 0: underflow error has occurred 1: underflow error has not underflowed, since last RSTGOFL assertion.
7	FLFF	R/O	Forward Link FIFO full flag 0: full 1: not full
6	FLPAF	R/O	Forward Link FIFO almost full flag 0: almost full 1: not almost full
5	FLPAE	R/O	Forward Link FIFO almost empty flag 0: almost empty 1: not almost empty
4	FLFE	R/O	Forward Link FIFO empty flag 0: empty 1: not empty
3	TEMPERATURE	R/O	This bit is the temperature bit. See the DS1620 data sheet for more details.
2	BLAST	R/O	i960 bus BLAST signal 0: This transfer is the last transfer in a burst 1: This transfer is not the last transfer in a burst.
1	DETECT	R/O	DB25 connector/break-out board detect 0: connector/break-out board detected 1: connector/break-out board not detected
0	LINT	R/O	Local interrupt from v96pbc chip 0: asserted 1: negated

3.3.5 PROGRAMMABLE FLAG /TEST DATA REGISTER

This register space can be used to adjust the programmable flags and send test data into the 4 FIFO banks.

To adjust a FIFO's programmable flag:

- (1) Determine which FIFO bank you want to adjust a FIFO's programmable flags:

For FIFO Bank #1 the FIFO-bank-constant is 0x0000E000.
 For FIFO Bank #2 the FIFO-bank-constant is 0x0000D000.
 For FIFO Bank #3 the FIFO-bank-constant is 0x0000B000.
 For FIFO Bank #4 the FIFO-bank-constant is 0x00007000.

- (2) Determine which FIFO's flags within the bank to adjust. There are 4 FIFO's per bank, numbered 0 - 3. This will be the FIFO number.

- (3) Build the 4 flag-adjust-bytes as per IDT72241 spec:

- pae_lsb
- pae_msb
- paf_lsb
- paf_msb

Refer to IDT72241 for the definition of these bytes.

- (4) Ensure that the FIFO to be adjusted is in the initial state. The easiest way to do this is to reset the FIFO bank. Note that this will reset all 4 FIFO's of the bank, including resetting their programmable flags back to the default value, so only do this once at the beginning if adjusting more than one flag in a bank.
- (5) Ensure that the stacker for the desired FIFO bank is in its initial state by checking the state bits in status register 2. Assert then negate RSTSTACKERS (via the control register) or write null data to cycle the stacker state machine to 00. Note that all four stackers are reset together when RSTSTACKERS is asserted.
- (6) Ensure that STACKORDER bit is low (via the board control register bit 27).
- (7) If the FIFO number is not 3, set the LD bit high (via the control register) for the desired FIFO bank and write (3 - FIFO number) bytes to the FIFO using the FIFO-bank-constant. For example to adjust the flags for FIFO number 1 in FIFO bank 3, perform

*0xXX280000 = 0x000B000;

3 - 1 = 2 times.

- (8) Set the LD bit low for the desired FIFO bank.

- (9) Write the appropriate flag-adjust-byte bitwise-or'ed with the FIFO-bank-constant. Using the same example, perform

$$*0xXX280000 = ((\text{ULONGWORD}) \text{flag-adjust-byte}) \mid 0x000B000;$$

- (10) Set the LD bit high for the desired FIFO bank and write 4 bytes to the FIFO using the FIFO-bank-constant to cycle the stacker back around to the desired FIFO. Using the same example, perform 4 times.

$$*0xXX280000 = 0x000B000;$$

- (11) Repeat steps 8- 10, using the next flag-adjust-byte in the sequence.
- (12) Repeat steps 5 - 11 to adjust another FIFO's flags in the same bank.
- (13) Flush out the extra data added to the FIFO's in steps 6 & 8 by setting LD high & repeatedly reading the selected FIFO bank until all of its FIFO's are empty.
- (14) The FIFO bank is now ready to receive real data and assert its flags at the programmed values. Set STACKORDER to the desired runtime value.

For example, to set the programmable flag for FIFO 2 of FIFO bank 3:

To verify a FIFO's programmable offset values:

- (1) Set LD for the desired FIFO bank low.
- (2) Read 4 longwords from the FIFO:

- The 1st longword will contain the pae_lsb's for each of the 4 FIFO's in the bank
- The 2nd longword will contain the pae_msb's for each of the 4 FIFO's in the bank
- The 3rd longword will contain the paf_lsb's for each of the 4 FIFO's in the bank
- The 4th longword will contain the paf_msb's for each of the 4 FIFO's in the bank

- (3) Set LD high (to put it at the runtime value)

To send test data into a FIFO bank:

- a) Determine which FIFO bank you want to test:

For FIFO Bank #1 the FIFO-bank-constant is 0x0000E000.
 For FIFO Bank #2 the FIFO-bank-constant is 0x0000D000.
 For FIFO Bank #3 the FIFO-bank-constant is 0x0000B000.
 For FIFO Bank #4 the FIFO-bank-constant is 0x00007000.

- b) Ensure that the FIFO is not in reset and it's LD bit is high.

c) Write the following:

$$*0xXX280000 = ((\text{ULONGWORD}) \text{byte_to_write}) |$$

$$(((\text{ULONGWORD}) \text{eof}) < 8) | \text{FIFO-bank-constant};$$

Where:

- byte_to_write is the next byte you wish to write into the FIFO
- eof is the end of frame bit, typically this is active-high. The stackers will use this bit, when high, in conjunction with STACKMODE to determine whether to pack or pad the frames into the FIFOs.

3.3.6 MISCELLANEOUS CONTROL REGISTER

Table 3-6. Miscellaneous Control Register

Reg	Name	Miscellaneous Control Register	
	Description	?	
	Address	\$xx30 0000	
	Port Size	longword (32 bits)	
	Reset Value	xxxx xxxx xxxx xxxx xxxx xxxx 0000 0000 (binary) XXXXXX00 (hex)	
Bit(s)	Name	R/W	Definition
31..7	-	-	Unused.
8	FLINTCTRL	R/W	Forward Link Interrupt Control 0: disabled 1: enabled
7..4	SOFTEV<3..0>	R/W	These bits are software definable bits which connect to testpoints, facilitating system performance measuring via a logic analyzer.
3	TEMPDQ	R/W	Temperature readout control data pin
2	TEMPDQZ	R/W	Temperature readout control 0: write control data into temperature chip 1: read temperature data out of the temperature chip
1	TEMPCLK	R/W	Temperature chip control clock
0	TEMPRST	R/W	Temperature chip reset 0: reset temperature chip 1: normal temperature chip operation

3.3.7 PIFS REGISTERS

Refer to §1.4, document 521-ASIC-023.

3.3.8 RS INTERNAL RAM

This is for testing and not runtime use. Refer to the RS documentation for more details.

3.3.9 RS REGISTERS

Refer to §1.4, document 521-SPEC-002.

3.3.10 RS ROUTING TABLE RAM

The routing table RAM is configured by software and used by the RS chip to determine how to route output frames.

The PCI10FR card uses routing bits 7..4 and 0 to select a FIFO bank.

If routing bit 0 is 1, the reject chip select will be output and the frame will be rejected, regardless of the other routing bits. The reject chip select can assert for multiple reasons.

If routing bit 7 is 1, the current frame will be routed to FIFO bank #1.

If routing bit 6 is 1, the current frame will be routed to FIFO bank #2.

If routing bit 5 is 1, the current frame will be routed to FIFO bank #3.

If routing bit 4 is 1, the current frame will be routed to FIFO bank #4.

More than 1 FIFO bank may be written to at a time.

See §1.4, document 521-SPEC-002, for more details.

3.3.11 FIFO #1

This is one of the 4 FIFO's in which processed telemetry data is buffered for transfer by the host. The RS chip uses the software-defined routing table to specify which of the 4 FIFO's (if any) to write data into.

3.3.12 FIFO #2

This is one of the 4 FIFO's in which processed telemetry data is buffered for transfer by the host. The RS chip uses the software-defined routing table to specify which of the 4 FIFO's (if any) to write data into.

3.3.13 FIFO #3

This is one of the 4 FIFO's in which processed telemetry data is buffered for transfer by the host. The RS chip uses the software-defined routing table to specify which of the 4 FIFO's (if any) to write data into.

3.3.14 FIFO #4

This is one of the 4 FIFO's in which processed telemetry data is buffered for transfer by the host. The RS chip uses the software-defined routing table to specify which of the 4 FIFO's (if any) to write data into.

3.3.15 FORWARD LINK FIFO

The Forward Link FIFO is a software writable device. A parallel to serial converter exists on the other side of this FIFO to serialize the input data. The data from the host must be de-stacked into bytes to write into this FIFO.

3.3.16 FLASH EEPROM

The Flash EEPROM is 512 Kbytes, organized as 128K x 32-bit, and is longword addressable. This memory provides for the evaluation/use of non-volatile memory. To write to this EEPROM you must follow the protocol defined for the part. Additionally, you must program in longword widths.

3.3.17 BOARD RAM

The Board RAM is 512 Kbytes of Static Ram (SRAM). This RAM is organized as 128K x 32-bit and is byte, word and longword addressable. This RAM provides for the evaluation and/or use of DMA chaining with the V96PBC chip.

SECTION 4 HARDWARE INSTALLATION

4.1 INTRODUCTION

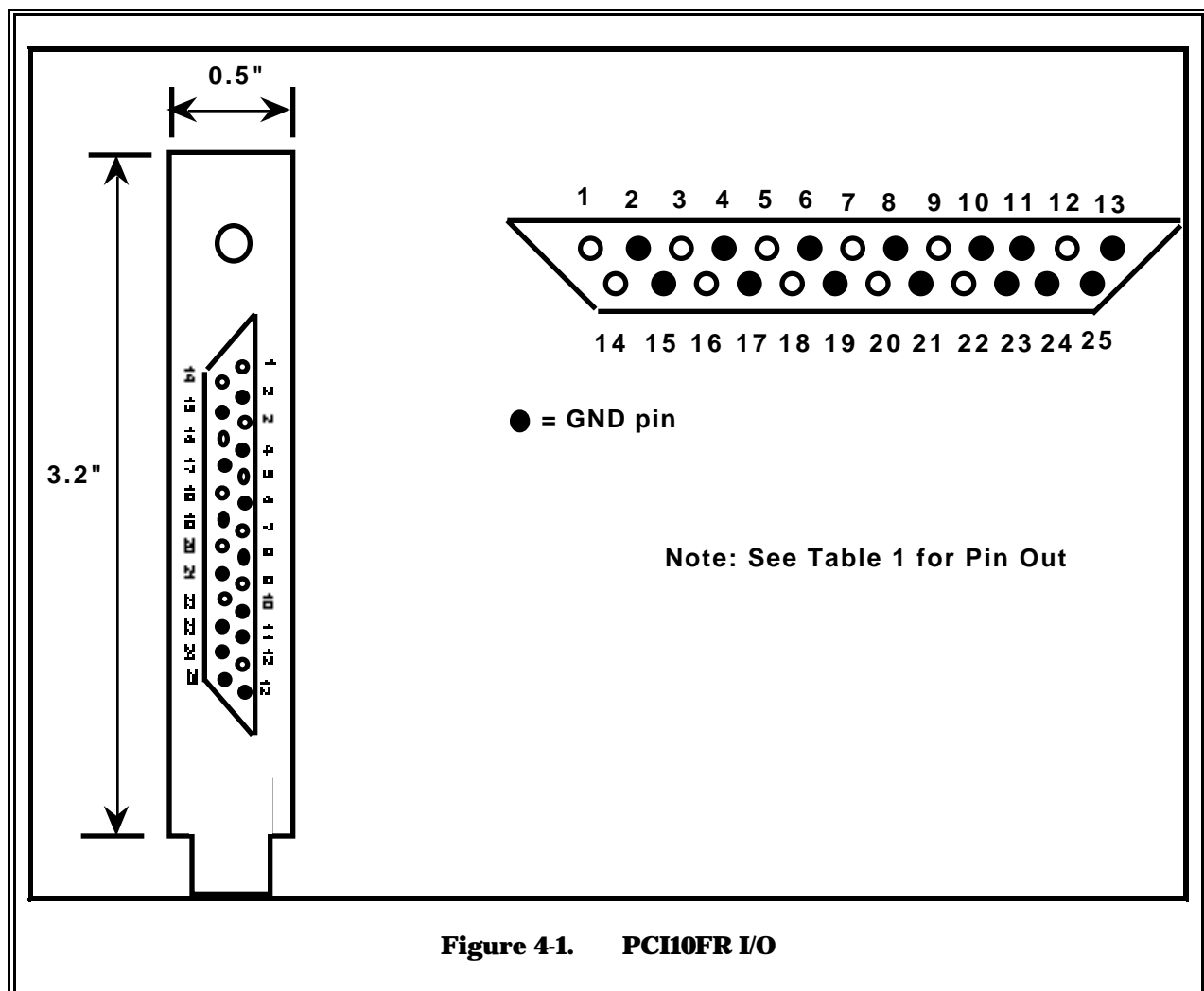
The PCI10FR has a total of two I/O interfaces: one on the I/O Panel which is generally accessible from the rear of the PCI chassis, and one not on the I/O Panel which is only accessible from inside the chassis.

This section provides an overview of the PCI10FR hardware and installation.

4.2 HARDWARE ELEMENTS

4.2.1 I/O PANEL

The I/O Panel interface operates through one LED and one input connector as depicted in Figure 4-1.

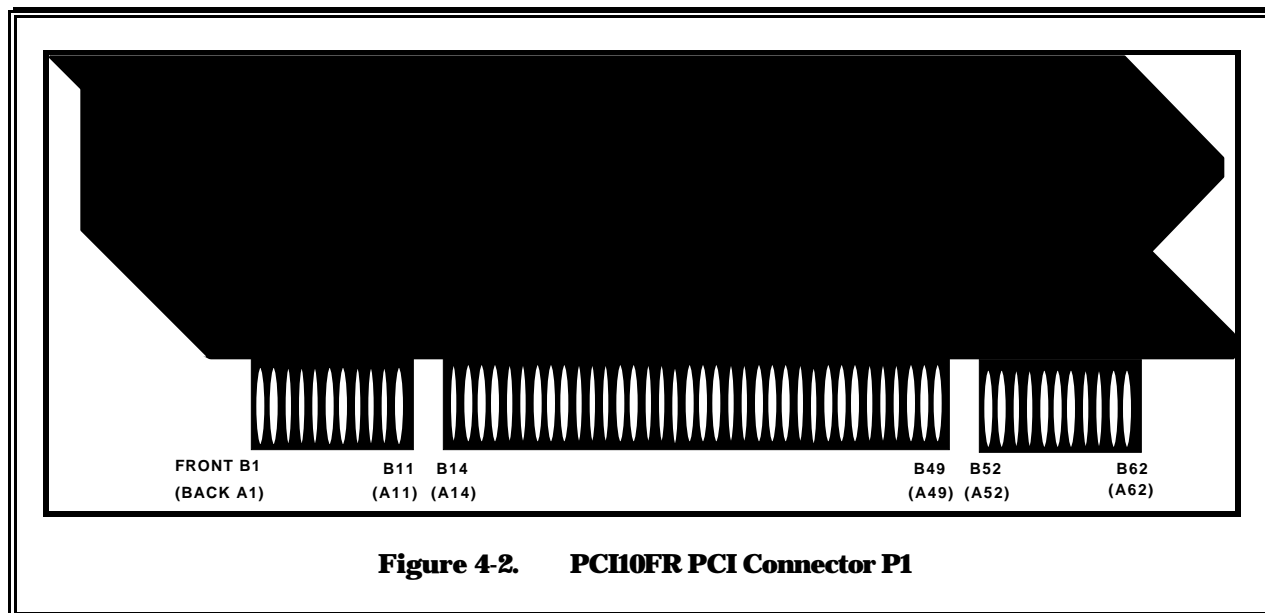


4.2.2 INTERNAL CONNECTORS

The PCI10FR contains one internal connector: the PCI connector. The placement of the internal connector appears in Figure 4-2.

4.2.2.1 PCI Connector P1

The PCI connector is a 124-pin double-sided board-edge plug as depicted in Figure 4-2.



4.2.3 JUMPERS AND SWITCHES

There are five switches connected in series with the load resistors across the RS422 inputs and RS422 outputs so that these I/Os can be used as part of a RS485 bus.

4.3 INTERFACES

The PCI10FR has a total of 4 interfaces: LED, RS-422 data input, RS-422 forward link output, and PCI.

4.3.1 LED

The I/O Panel contains one green LED (D1); see Figures 4-1 and 4-2 and Table 4-1. Its operation is entirely under software control; writing a 0 to bit 16 of the Board Control register \$00 000 turns it on, and writing a 1 turns it off. It can be useful for testing, condition indication, "heartbeat" indication, or simple power. Upon reset it defaults to "on."

Table 4-1. PCI10FR I/O Panel Interfaces

Ref	Name	Description		
D1	LED	LED, controlled by the host (software), green		
J1		DB-25 jack, low-frequency (10 MHz) input signals:		
		Pin	Signal	Description
		14	RXD+	RS-422 positive receiving data input, 120W impedance switchable
		1	RXD-	RS-422 negative receiving data input, 120W impedance switchable
		16	RXC+	RS-422 positive receiving clock input, 120W impedance switchable
		3	RXC-	RS-422 negative receiving clock input, 120W impedance switchable
		18	TXD+	RS-422 positive forward link data output, 120W impedance switchable
		5	TXD-	RS-422 negative forward link data output, 120W impedance switchable
		20	TXC+	RS-422 positive forward link clock output, 120W impedance switchable
		7	TXC-	RS-422 negative forward link clock output, 120W impedance switchable
		22	FLCLK+	RS-422 positive forward link external synchronization clock input, 120W impedance switchable
		9	FLCLK-	RS-422 negative forward link external synchronization clock input, 120W impedance switchable
		12	DETECT	+5V TTL input signal with 6KW pullup
		2,4,6,8,10,11,13,15,17,19,21,23,24,25	GND	signal ground

4.3.2 RS-422 DATA INPUT INTERFACE

The I/O Panel contains one DB-25 jack (J1) for low-frequency input signals (ILF); see Figures 4-1 and Table 4-1. It contains pins which will accept RS-422 type satellite telemetry data signals at up to 10 Mbps nominal, sometimes higher. The inputs are: positive data (pin 14, RXD+), negative data (pin 1, RXD-), positive clock (pin 16, RXC+), and negative clock (pin 3, RXC-). They are terminated with switchable 120W resistors to their differential counterparts, so 120W shielded twisted-pair cable should be used. Low-rate data enters the PIFS through its bit-serial data input port number 0, so it must be set up appropriately. Low-rate inputs left connected when unused have no effect.

4.3.3 RS-422 FORWARD LINK DATA OUTPUT INTERFACE

J1 also contains pins which will output RS-422 type satellite forward link data signals at up to 8 Mbps in 1/2 Hz steps. The outputs are: positive data (pin 18, TXD+), negative data (pin 5, TXD-), positive clock (pin 20, TXC+), and negative clock (pin 7, TXC-). The inputs are: external synchronization positive clock (pin22, FLCLK+), external synchronization negative clock (pin9, FLCLK-), and TTL sense detect (pin12, DETECT).

4.3.4 PCI BUS INTERFACE

The Return Link Processor (RLP) interfaces directly with a host computer through the industry-standard PCI bus connector P1; see Figures 4-2 and Table 4-2. Further information is available from the reference documents listed in §1.4 and the PCI10FR schematics. The Return Link Processor (RLP) interfaces directly with a host computer through the industry-standard PCI bus connector P1; see Figures 4-2 and Table 4-2.

Table 4-2. PCI10FR PCI Bus Connector P1 Signals

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
TRST	A1	AD16	A32	VN12_0	B1	AD17	B32
VP12_0	A2	VDD	A33	TCK	B2	CBE2	B33
TMS	A3	FRAME	A34	GND	B3	GND	B34
TDI	A4	GND	A35	TDO	B4	IRDY	B35
VP5_0	A5	TRDY	A36	VP5_0	B5	VDD	B36
INTA	A6	GND	A37	VP5_0	B6	DEVSEL	B37
INTC	A7	STOP	A38	INTB	B7	GND	B38
VP5_0	A8	VDD	A39	INTD	B8	LOCK	B39
NC	A9	SDONE	A40	PRSNT1	B9	PERR	B40
NC	A10	SBO	A41	NC	B10	VDD	B41
NC	A11	GND	A42	PRSNT2	B11	SERR	B42
KEYWAY	A12	PAR	A43	KEYWAY	B12	VDD	B43
KEYWAY	A13	AD15	A44	KEYWAY	B13	CBE1	B44
NC	A14	VDD	A45	NC	B14	AD14	B45
RST	A15	AD13	A46	GND	B15	GND	B46
NC	A16	AD11	A47	CLK	B16	AD12	B47
GNT	A17	GND	A48	GND	B17	AD10	B48
GND	A18	AD9	A49	REQ	B18	M66EN	B49
NC	A19	KEYWAY	A50	NC	B19	KEYWAY	B50
AD30	A20	KEYWAY	A51	AD31	B20	KEYWAY	B51
VDD	A21	CBE0	A52	AD29	B21	AD8	B52
AD28	A22	VDD	A53	GND	B22	AD7	B53
AD26	A23	AD6	A54	AD27	B23	VDD	B54
GND	A24	AD4	A55	AD25	B24	AD5	B55
AD24	A25	GND	A56	VDD	B25	AD3	B56
IDSEL	A26	AD2	A57	CBE3	B26	GND	B57
VDD	A27	AD0	A58	AD23	B27	AD1	B58
AD22	A28	NC	A59	GND	B28	NC	B59
AD20	A29	REQ64	A60	AD21	B29	ACK64	B60
GND	A30	VP5_0	A61	AD19	B30	VP5_0	B61
AD18	A31	VP5_0	A62	VDD	B31	VP5_0	B62

4.4 INSTALLATION GUIDELINES

Follow the standard procedures for installing a PCI expansion card in a PCI host chassis. In brief, this involves having the power off but not unplugged (maintaining the integrity of the ground), properly grounding the installer and surroundings, opening the chassis, removing an I/O Panel filler plate if necessary, inserting the PCI10FR, fastening the I/O Panel screw, closing the chassis, and finally connecting all appropriate input cables. The driver and application software must also be installed.

4.5 ENVIRONMENTAL REQUIREMENTS

The PCI10FR is a standard commercial-grade computer product. It will operate from 0 to +70 degrees Celsius. It is not designed to withstand any unusual levels of physical shock or nuclear or electromagnetic radiation. It is electrostatic discharge (ESD)-sensitive and all appropriate precautions must be taken before and during its handling. It is only specified to operate in a PCI Revision 2.1-compliant host platform (refer to §1.4).

SECTION 5 OPERATING PRINCIPLES

5.1 INTRODUCTION

During normal operation, the PCI10FR is basically seen by the satellite telemetry bitstream provider as a data sink, and by the PCI host computer as a PCI data source. Since the PCI10FR has no MPU of its own, it behaves as a simple data flow device with straightforward operation: set the card up prior to a data session, and service it (move data off the card as it becomes available) during the session. The most critical aspect is in servicing the card fast enough and efficiently enough to avoid data loss.

This section outlines procedures to set up, control, and monitor the PCI10FR and the flow of data.

5.2 PRELIMINARIES

For the PCI10FR to function, the serial-EEPROM and the seven PLDs must be programmed. In particular, the EEPROM must be programmed out-of-circuit prior to board assembly. While it is re-programmable in-circuit via the PCI bus, the board must first function on the bus to make reprogramming possible. But because PCI configuration is read from the EEPROM by the PBC, this cannot happen without valid content. The PLDs can be programmed out-of-circuit prior to assembly. In all cases, device reprogramming is a development function only and should not be encountered during normal end-user operation; further information is available from the documents listed in §1.4 .

5.3 SETUP

Upon power-up or reset of the PCI host computer, the PCI10FR's PBC automatically reads PCI configuration data from its on-board ROM and sets up its PCI interface. Then depending on the host, either the operating system (OS) or the driver software reads the PCI10FR's PCI configuration and initializes itself accordingly. At this point the PCI10FR is ready to be set up for a data session by the application software.

5.3.1 RESETS AND INPUTS

After power-up or reset of the PCI host computer, most of the PCI10FR circuits are held in reset by the Baseboard Interface. The only circuits that are accessible in this condition are the DMA-chaining memory, The EEPROM flash memory, the Baseboard Interface, and the PBC. All the resets can be released at this point through the Board Control register. Activity on the RS-422 input has no effect. Refer to §3.3.1 and the schematics for more details.

5.3.2 PCI INTERFACE

The PBC must be set up. Local bus byte enables are not used (always enabled) and local bus parity checking is not used. The PBC operates at the same frequency of 33 MHz as the local bus. It inputs the interrupt from the Baseboard Interface on INTD, outputs the interrupt to the PCI bus on INTA, and does not use INTB (which is connected to the PCI bus). Refer to the documents listed in §1.4 and the schematics for more details.

5.3.3 INPUT INTERFACE

Input terminal resistors shall be switched on or switched off accordingly. Refer to §6 and the schematics for more details.

5.3.4 PIFS

The PIFS must be set up according to user and data session requirements. If the input source is RS-422, use the PIFS serial input port number 0. Data can also be injected by the host through the forward link and loopback into the PIFS serial input port number 1. Data is output from the PIFS 16 bits at a time; 16 is normally suggested for optimal performance when outputting to the RS. The internal timecode reference clock is 10 MHz, and the processing clock is 33 MHz. Refer to the documents listed in §1.4 and §6 for details.

5.3.5 DATA ROUTING MODE

Data may be routed through the PCI10FR in one way: it goes from the PIFS through the RS then into the Channel Routing Data FIFOs. Refer to §6 for more details.

5.3.6 RS

The RS must be set up according to user and data session requirements; it has forty 16-bit registers appearing in 40 contiguous 32-bit locations, one register per location. The input data size of 8 or 16 bits must match the output data size set up in the PIFS. Data output from the RS can be either 8 or 16 bits wide; 16 is normally suggested for optimal performance. The RS has four external Channel Routing FIFOs for routing data to multiple channels. It runs off the 33 MHz processing clock. Refer to the documents listed in §1.4 and the schematics in §6 for more details.

5.3.7 INTERRUPTS

The same conditions, signals, and registers used to set up, control, and monitor interrupts are also used for polling, the only difference being that for polling, the assertion of the baseboard interrupt signal is disabled. Therefore where polling is not explicitly mentioned in the discussion, it is implied.

The interrupt source masks must be set up. These reside in Baseboard Interface Interrupt Source Mask bits. If a source is masked it will not be used to generate the baseboard interrupt. Upon power-up or reset of the PCI host computer, all interrupt sources default to masked. Refer to §3.3.2 for more details.

5.3.8 PROGRAMMABLE FIFO FLAGS

All channel routing data FIFOs have programmable almost-full and almost-empty flags which can be reprogrammed differently from their defaults. Refer to the documents listed in §1.4 for the basic default and programming information. To program any of these FIFOs, the Universal FIFO Programmable Flag Load bit of the Baseboard Interface Control register must be set to 0 or “load” while programming data is being written. At all other times it must be set to 1 or “normal”. After power-up or reset of the PCI host computer, this bit defaults to “load”. Refer to §6 for more details.

The channel routing Data FIFOs inputs are multiplexed to route data either from RS or from local bus. To program the programmable flags of these FIFOs, the multiplexer (MUX) is switched to route local bus data into the inputs of the FIFOs and program them according to the specification of the FIFO data sheet.

5.3.9 BASEBOARD INTERFACE

The only setup for the Baseboard Interface involves the reset, input, routing, and interrupt functions discussed in §5.3.1, §5.3.3, §5.3.5, and §5.3.7, respectively.

5.4 OPERATION

When the setup is complete, the PIFS inputs are enabled, the baseboard interrupt is enabled (if used), and the PCI10FR is ready to begin a data session. The typical service cycle consists of four phases: waiting for an accumulation of data, initiating a data transfer, waiting for completion of the DMA, and then rechecking for more data. At the conclusion of a session, any data remaining in-process may be flushed. If interrupt latency is found to limit performance unacceptably, polling must be used when waiting for data; whether or not polling is superior to interrupts for detecting when a DMA is complete is system-dependent and must be determined empirically. The same conditions, signals, and registers used to set up, control, and monitor interrupts are also used for polling, the only difference being that for polling, the assertion of the board interrupt signal is disabled. Therefore, where polling is not explicitly mentioned in the discussion, it is implied.

5.4.1 WAIT FOR DATA

While no DMA is in progress, software waits for enough data to accumulate in one or more of the four 32-bit Channel Routing FIFO banks. It may wait for an interrupt or may poll Status Register and Status Register 2. During this phase it may also toggle the LED by changing the Board Control Register LED bit.

5.4.2 INITIATE A TRANSFER

Upon receiving an interrupt, the driver software's interrupt service routine (ISR) first checks the V962PBC chip's PCI Interrupt Status Register to make sure that the interrupt originated from the board. If the interrupt is not from the board, the ISR returns quickly and allows the interrupt to be serviced by other drivers' ISR's. If the interrupt did originate on the board, the ISR determines whether the interrupt is due to completion of a DMA operation or due to other causes on the board.

If the interrupt was not due to completion of a DMA operation, then Status Registers 1 and 2 are read to determine which of the on-board sources of interrupt occurred. If one of the four FIFO banks generated the interrupt due to the almost-full flag being asserted, the ISR disables the FIFO's interrupt, initiates a DMA operations to move data from the FIFO into host memory, clears the V962PBC chip's PCI Interrupt Status Register, and returns from the interrupt. On a Windows NT platform, the ISR actually calls a deferred procedure call (DPC) to initialize the DMA operation, but otherwise the routines are the same. Since any number of the FIFO's may need servicing at a given time, the ISR checks all of them whether they generated the interrupt or not. This makes the ISR much more efficient.

5.4.3 WAIT FOR DMA COMPLETION

A DMA transaction begins as soon as it is initiated. While a DMA transaction is ongoing, PCI bus traffic must be kept to an absolute minimum to ensure optimum performance. This means the host should not attempt to access the PCI10FR at all until the DMA is complete and it receives the DMA-Done interrupt. That is, unless it is polling the PBC to ascertain this condition. It is unclear which has the larger impact, interrupt latency of the DMA-Done interrupt or bus interference of polling; this must be experimentally-determined on a platform-by-platform basis.

During this time the host is free to perform local processing, although it will most likely be called upon to transfer data already in host memory to an I/O device for storage.

5.4.4 RECHECK FOR DATA

When the DMA operation has completed, an interrupt will be generated and the ISR will once again determine the cause. If the interrupt is due to a DMA operation completing, the ISR will clear the DMA interrupt through the V962PBC chip's PCI Interrupt Status Register and the Local Bus Interrupt Control and Status Register. On a Windows NT platform, a DPC will be queued to update memory pointers, asynchronously notify the user application that data has been successfully transferred into host memory, and to reenable the FIFO's interrupt. Other platforms will perform these operations within the ISR. Again, all sources of interrupt are checked and appropriate action is taken. The driver then goes back into a "Waiting for Data" phase.

5.4.5 FLUSH REMAINING DATA

Eventually after a data session has terminated, all of the interrupt sources have been serviced and none remain. At this point, some data may still exist in the FIFO's. A flush operation may be performed to capture this data.

The flush operation must first turn the PIFS chip off so that no new data gets into the pipeline. The PIFS and RS chips will automatically timeout and move their data in the data FIFO's, if these options are turned on. The flush operation basically performs a programmed I/O operation to read data from the FIFO one 32-bit long word at a time. The operation polls the empty flag of the specific FIFO to determine when all of the data has been downloaded.

SECTION 6 SCHEMATIC DESCRIPTION

6.1 INTRODUCTION

In addition to the customary two assembly and one drill drawings, the PCI10FR schematic has 14 pages. All drawings are provided in Appendix A. There are also seven PLDs and one ROM, and their programs are listed in Appendix B. This section describes the drawings and program listings.

6.2 DRAWING DIRECTORY

The PCI10FR drawings are listed in Table 6-1.

Table 6-1. PCI10FR Drawing Directory

Sheet	Title
-	Top Assembly
-	Bottom Assembly
-	Drill Master
1	Table of Contents
2	RS422 Input, PIFS Chip and PIFS-RS FIFOs
3	RS Chip, RS Output Buffer and Programmable Flag Mux
4	RSEDC Routing Table Logic
5	RS-DMA FIFO Banks 1 & 2
6	RS-DMA FIFO Banks 3 & 4
7	Forward-Link Subsystem 1 of 2, Clock Generation
8	Forward-Link Subsystem 1 of 2, P2S Logic
9	Flash EEPROM and SRAM
10	Control and Status Registers
11	CS Generation, Bridge Chip, and PCI & Back-Panel I/O Connectors
12	Decoupling Caps, Pull-Ups, Terminators and VCC Mapping
13	Test Points, Interrupt and Miscellaneous Logic
14	Additional Pull-Ups

6.3 DRAWING DESCRIPTION

6.3.1 TOP ASSEMBLY

Used by assembly to place the devices on the front side of the PCI10FR printed circuit board.

6.3.2 BOTTOM ASSEMBLY

Used by assembly to place the devices on the back side of the PCI10FR printed circuit board.

ACRONYMS AND ABBREVIATIONS

Term	Definition
AOS	Advanced Orbiting Systems
ASIC	Application-Specific Integrated Circuit
BTD	Bit Transition Density
CCSDS	Consultative Committee on Space Data Systems
CMOS	Complimentary Metal-Oxide-Semiconductor
COTS	Commercial Off-the-Shelf
CRC	Cyclic Redundancy Code
DEC	Digital Equipment Corporation
DMA	Direct Memory Access
DPC	Deferred Procedure Call
DSDP	Desktop Satellite Data Processor
DSTD	Data Systems Technology Division
EDC	Error Detector/Corrector
EEPROM	Electrically-Erasable PROM
EPROM	Erasable PROM
ESD	Electrostatic Discharge
FIFO	First-In, First-Out
FS	Frame Synchronizer
GaAs	Gallium-Arsenide
GSFC	Goddard Space Flight Center
IBM	International Business Machines
I/O	Input/Output
ISR	Interrupt Service Routine
LAN	Local Area Network
LED	Light-Emitting Diode
MO&DSD	Mission Operations and Data Systems Directorate
MPU	Microprocessing Unit
MSB	Microelectronic Systems Branch
MUX	Multiplexer
NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
NGS	Next Generation Systems
NRZ	Non-Return-to-Zero
OS	Operating System
P&P	Plug-and-Play
PBC	PCI Bridge Chip
PCI	Peripheral Components Interface

PIFS	Parallel, Integrated FS
PLD	Programmable Logic Device
PROM	Programmable ROM
RAM	Random-Access Memory
ROM	Read-Only Memory
RS	Reed-Solomon
SRAM	Static RAM
TTL	Transistor-Transistor Logic
VME	Versa-Module Euroboard
WAN	Wide Area Network

6.3.3 DRILL MASTER

Used by fabrication to cut and drill the PCI10FR printed circuit board.

6.3.4 SHEET 1

PCI10FR schematic table of contents, list of design revisions.

6.3.5 SHEET 2

DB-25 (RS-422) input circuit, PIFS ASIC, PIFS output FIFO, internal 10 MHz timecode reference clock oscillator, 33 MHz processing clock oscillator and distribution circuit.

6.3.6 SHEET 3

RS ASIC, RS output buffer and programmable flag MUX.

6.3.7 SHEET 4

RS routing table SRAM, routing table address selection MUX.

6.3.8 SHEET 5

RS-DMA channel routing FIFO banks 1 & 2 and control.

6.3.9 SHEET 6

RS-DMA channel routing FIFO banks 3 & 4 and control.

6.3.10 SHEET 7

Forward link output clock selection and programmable clock generation.

6.3.11 SHEET 8

Forward link byte wide FIFO, forward link byte parallel to bit serial controller, forward link RS422 output interface.

6.3.12 SHEET 9

Flash EEPROM and SRAM.

6.3.13 SHEET 10

Status registers, board control register.

6.3.14 SHEET 11

Local bus control logic, V962PBC local bus to PCI interface chip, V962PBC serial powerup setup EEPROM, PCI bus connector, back-panel connector, LED.

6.3.15 SHEET 12

+5V bypass capacitors, +5V pull-up resistors, clock terminators.

6.3.16 SHEET 13

Test points for debugging, temperature sensor interface, miscellaneous control register, interrupt signal generation control.

6.3.17 SHEET 14

additional pull-up resistors, 3.3V bypass capacitors, current sensing resistor and test points, additional miscellaneous control register bits.

6.4 PROGRAM LISTING DIRECTORY

The PCI10FR programmable device program listings are listed in Table 6-2.

Table 6-2. PCI10FR Program Listing Directory

ITEM	Description
1	Baseboard Interface Program (cpusigs.src)
2	Forward Link Interface Program (p2s.src)
3	Channel Routing Stack Ordering Program (stacker.src)
4	Interrupt Generation Program (int.src)
5	PCI Configuration ROM Data

6.4.1 BASEBOARD INTERFACE PROGRAM (CPUSIGS.SRC)

The Baseboard Interface firmware consists of three Lattice GAL26CV12-7 PLDs ; see the references listed in §1.4 for device information.

6.4.2 FORWARD LINK INTERFACE PROGRAM (P2S.SRC)

The Forward Link Interface firmware consists of a Lattice GAL26CV12-7 PLDs ; see the references listed in §1.4 for device information.

6.4.3 CHANNEL ROUTING STACK ORDERING PROGRAM (STACKER.SRC)

The Forward Link Stack Ordering firmware consists of two identical Lattice GAL26CV12-7 PLDs ; see the references listed in §1.4 for device information.

6.4.4 INTERRUPT GENERATION PROGRAM (INT.SRC)

The Interrupt Generation firmware consists of a Lattice GAL26CV12-7 PLDs ; see the references listed in §1.4 for device information.

6.4.5 PCI CONFIGURATION ROM

The PCI Configuration ROM is an ATMEL AT24C02 serial EEPROM; see §1.4 for device information.

It contains the first 128 bytes of PCI configuration space that the PBC reads to set its PCI Configuration values after power-up or reset. See the references listed in §1.4 for details.